

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Reading the CSV file

```
In [2]: df=pd.read_csv('zomato.csv')
df.head()
```

Out[2]:

	url	address	name	online_order	book_table	rate	votes	phone	location
0	https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1/5	775	42297555\r\n+91 9743772233	Banashankari
1	https://www.zomato.com/bangalore/spice-elephan...	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1/5	787	080 41714161	Banashankari
2	https://www.zomato.com/SanchurroBangalore?cont...	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	No	3.8/5	918	+91 9663487993	Banashankari
3	https://www.zomato.com/bangalore/addhuri-udupi...	1st Floor, Annakuteera, 3rd Stage, Banashankar...	Addhuri Udupi Bhojana	No	No	3.7/5	88	+91 9620009302	Banashankari
4	https://www.zomato.com/bangalore/grand-village...	10, 3rd Floor, Lakshmi Associates, Gandhi Baza...	Grand Village	No	No	3.8/5	166	8026612447\r\n+91 9901210005	Basavanagudi

```
In [3]: df.shape
```

Out[3]: (51717, 17)

```
In [4]: df.columns
```

Out[4]: Index(['url', 'address', 'name', 'online_order', 'book_table', 'rate', 'votes', 'phone', 'location', 'rest_type', 'dish_liked', 'cuisines', 'approx_cost(for two people)', 'reviews_list', 'menu_item', 'listed_in(type)', 'listed_in(city)'], dtype='object')

```
In [5]: df=df.drop(['url', 'address', 'phone', 'dish_liked','reviews_list', 'menu_item'],axis=1)
df.head()
```

Out[5]:

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	approx_cost(for two people)	listed_in(type)	listed_in(city)
0	Jalsa	Yes	Yes	4.1/5	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	800	Buffet	Banashankari
1	Spice Elephant	Yes	No	4.1/5	787	Banashankari	Casual Dining	Chinese, North Indian, Thai	800	Buffet	Banashankari
2	San Churro Cafe	Yes	No	3.8/5	918	Banashankari	Cafe, Casual Dining	Cafe, Mexican, Italian	800	Buffet	Banashankari
3	Addhuri Udupi Bhojana	No	No	3.7/5	88	Banashankari	Quick Bites	South Indian, North Indian	300	Buffet	Banashankari
4	Grand Village	No	No	3.8/5	166	Basavanagudi	Casual Dining	North Indian, Rajasthani	600	Buffet	Banashankari

```
In [6]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51717 entries, 0 to 51716
Data columns (total 11 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   name                                51717 non-null  object
1   online_order                        51717 non-null  object
2   book_table                          51717 non-null  object
3   rate                                43942 non-null  object
4   votes                              51717 non-null  int64
5   location                            51696 non-null  object
6   rest_type                          51490 non-null  object
7   cuisines                           51672 non-null  object
8   approx_cost(for two people)        51371 non-null  object
9   listed_in(type)                    51717 non-null  object
10  listed_in(city)                    51717 non-null  object
dtypes: int64(1), object(10)
memory usage: 4.3+ MB
```

Dropping Duplicate values

```
In [7]: df.drop_duplicates(inplace=True)
df.shape
```

```
Out[7]: (51609, 11)
```

```
In [8]: df['rate'].unique()
```

```
Out[8]: array(['4.1/5', '3.8/5', '3.7/5', '3.6/5', '4.6/5', '4.0/5', '4.2/5',
               '3.9/5', '3.1/5', '3.0/5', '3.2/5', '3.3/5', '2.8/5', '4.4/5',
               '4.3/5', 'NEW', '2.9/5', '3.5/5', nan, '2.6/5', '3.8 /5', '3.4/5',
               '4.5/5', '2.5/5', '2.7/5', '4.7/5', '2.4/5', '2.2/5', '2.3/5',
               '3.4 /5', '-', '3.6 /5', '4.8/5', '3.9 /5', '4.2 /5', '4.0 /5',
               '4.1 /5', '3.7 /5', '3.1 /5', '2.9 /5', '3.3 /5', '2.8 /5',
               '3.5 /5', '2.7 /5', '2.5 /5', '3.2 /5', '2.6 /5', '4.5 /5',
               '4.3 /5', '4.4 /5', '4.9/5', '2.1/5', '2.0/5', '1.8/5', '4.6 /5',
               '4.9 /5', '3.0 /5', '4.8 /5', '2.3 /5', '4.7 /5', '2.4 /5',
               '2.1 /5', '2.2 /5', '2.0 /5', '1.8 /5'], dtype=object)
```

Removing the 'NEW', '-' and '/5' from Rate column

```
In [9]: def handleRate(value):
         if(value=='NEW' or value=='-'):
             return np.nan
         else:
             value=str(value).split('/')
             value=value[0]
             return float(value)

df['rate']=df['rate'].apply(handleRate)
df['rate'].head()
```

```
Out[9]: 0    4.1
        1    4.1
        2    3.8
        3    3.7
        4    3.8
        Name: rate, dtype: float64
```

```
In [10]: df.rate.isnull().sum()
```

```
Out[10]: 10019
```

Filling NULL values in rate column with mean

```
In [11]: df['rate'].fillna(df['rate'].mean(),inplace=True)
df['rate'].isnull().sum()
```

```
Out[11]: 0
```

In [12]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
Index: 51609 entries, 0 to 51716
Data columns (total 11 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   name                                  51609 non-null  object
1   online_order                         51609 non-null  object
2   book_table                           51609 non-null  object
3   rate                                 51609 non-null  float64
4   votes                                51609 non-null  int64
5   location                             51588 non-null  object
6   rest_type                           51382 non-null  object
7   cuisines                             51564 non-null  object
8   approx_cost(for two people)         51265 non-null  object
9   listed_in(type)                     51609 non-null  object
10  listed_in(city)                     51609 non-null  object
dtypes: float64(1), int64(1), object(9)
memory usage: 4.7+ MB
```

Dropping NULL Values

In [13]: df.dropna(inplace=True)
df.head()

Out[13]:

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	approx_cost(for two people)	listed_in(type)	listed_in(city)
0	Jalsa	Yes	Yes	4.1	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	800	Buffet	Banashankari
1	Spice Elephant	Yes	No	4.1	787	Banashankari	Casual Dining	Chinese, North Indian, Thai	800	Buffet	Banashankari
2	San Churro Cafe	Yes	No	3.8	918	Banashankari	Cafe, Casual Dining	Cafe, Mexican, Italian	800	Buffet	Banashankari
3	Addhuri Udupi Bhojana	No	No	3.7	88	Banashankari	Quick Bites	South Indian, North Indian	300	Buffet	Banashankari
4	Grand Village	No	No	3.8	166	Basavanagudi	Casual Dining	North Indian, Rajasthani	600	Buffet	Banashankari

In [14]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
Index: 51042 entries, 0 to 51716
Data columns (total 11 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   name                                  51042 non-null  object
1   online_order                         51042 non-null  object
2   book_table                           51042 non-null  object
3   rate                                 51042 non-null  float64
4   votes                                51042 non-null  int64
5   location                             51042 non-null  object
6   rest_type                           51042 non-null  object
7   cuisines                             51042 non-null  object
8   approx_cost(for two people)         51042 non-null  object
9   listed_in(type)                     51042 non-null  object
10  listed_in(city)                     51042 non-null  object
dtypes: float64(1), int64(1), object(9)
memory usage: 4.7+ MB
```

Renaming the column

```
In [15]: df.rename(columns={'approx_cost(for two people)': 'CostFor2Plates', 'listed_in(type)': 'Type', 'listed_in(city)': 'City'},
df.head())
```

Out[15]:

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	CostFor2Plates	Type	City
0	Jalsa	Yes	Yes	4.1	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	800	Buffet	Banashankari
1	Spice Elephant	Yes	No	4.1	787	Banashankari	Casual Dining	Chinese, North Indian, Thai	800	Buffet	Banashankari
2	San Churro Cafe	Yes	No	3.8	918	Banashankari	Cafe, Casual Dining	Cafe, Mexican, Italian	800	Buffet	Banashankari
3	Addhuri Udupi Bhojana	No	No	3.7	88	Banashankari	Quick Bites	South Indian, North Indian	300	Buffet	Banashankari
4	Grand Village	No	No	3.8	166	Basavanagudi	Casual Dining	North Indian, Rajasthani	600	Buffet	Banashankari

```
In [16]: df['location'].unique()
```

```
Out[16]: array(['Banashankari', 'Basavanagudi', 'Mysore Road', 'Jayanagar',
'Kumaraswamy Layout', 'Rajarajeshwari Nagar', 'Vijay Nagar',
'Uttarahalli', 'JP Nagar', 'South Bangalore', 'City Market',
'Nagarbhavi', 'Bannerghatta Road', 'BTM', 'Kanakapura Road',
'Bommanahalli', 'CV Raman Nagar', 'Electronic City', 'HSR',
'Marathahalli', 'Wilson Garden', 'Shanti Nagar',
'Koramangala 5th Block', 'Koramangala 8th Block', 'Richmond Road',
'Koramangala 7th Block', 'Jalahalli', 'Koramangala 4th Block',
'Bellandur', 'Sarjapur Road', 'Whitefield', 'East Bangalore',
'Old Airport Road', 'Indiranagar', 'Koramangala 1st Block',
'Frazer Town', 'RT Nagar', 'MG Road', 'Brigade Road',
'Lavelle Road', 'Church Street', 'Ulsoor', 'Residency Road',
'Shivajinagar', 'Infantry Road', 'St. Marks Road',
'Cunningham Road', 'Race Course Road', 'Commercial Street',
'Vasanth Nagar', 'HBR Layout', 'Domlur', 'Ejipura',
'Jeevan Bhima Nagar', 'Old Madras Road', 'Malleshwaram',
'Seshadripuram', 'Kammanahalli', 'Koramangala 6th Block',
'Majestic', 'Langford Town', 'Central Bangalore', 'Sanjay Nagar',
'Brookefield', 'ITPL Main Road, Whitefield',
'Varthur Main Road, Whitefield', 'KR Puram',
'Koramangala 2nd Block', 'Koramangala 3rd Block', 'Koramangala',
'Hosur Road', 'Rajajinagar', 'Banaswadi', 'North Bangalore',
'Nagawara', 'Hennur', 'Kalyan Nagar', 'New BEL Road', 'Jakkur',
'Rammurthy Nagar', 'Thippasandra', 'Kaggadasapura', 'Hebbal',
'Kengeri', 'Sankey Road', 'Sadashiv Nagar', 'Basaveshwara Nagar',
'Yeshwantpur', 'West Bangalore', 'Magadi Road', 'Yelahanka',
'Sahakara Nagar', 'Peenya'], dtype=object)
```

```
In [17]: df['City'].unique()
```

```
Out[17]: array(['Banashankari', 'Bannerghatta Road', 'Basavanagudi', 'Bellandur',
'Brigade Road', 'Brookefield', 'BTM', 'Church Street',
'Electronic City', 'Frazer Town', 'HSR', 'Indiranagar',
'Jayanagar', 'JP Nagar', 'Kalyan Nagar', 'Kammanahalli',
'Koramangala 4th Block', 'Koramangala 5th Block',
'Koramangala 6th Block', 'Koramangala 7th Block', 'Lavelle Road',
'Malleshwaram', 'Marathahalli', 'MG Road', 'New BEL Road',
'Old Airport Road', 'Rajajinagar', 'Residency Road',
'Sarjapur Road', 'Whitefield'], dtype=object)
```

```
In [18]: df=df.drop(['City'],axis=1)
```

```
In [19]: df['CostFor2Plates'].unique()
```

```
Out[19]: array(['800', '300', '600', '700', '550', '500', '450', '650', '400',
'900', '200', '750', '150', '850', '100', '1,200', '350', '250',
'950', '1,000', '1,500', '1,300', '199', '80', '1,100', '160',
'1,600', '230', '130', '50', '190', '1,700', '1,400', '180',
'1,350', '2,200', '2,000', '1,800', '1,900', '330', '2,500',
'2,100', '3,000', '2,800', '3,400', '40', '1,250', '3,500',
'4,000', '2,400', '2,600', '120', '1,450', '469', '70', '3,200',
'60', '560', '240', '360', '6,000', '1,050', '2,300', '4,100',
'5,000', '3,700', '1,650', '2,700', '4,500', '140'], dtype=object)
```

Removing Comma(,) from the CostFor2Plates column

```
In [20]: def handlecomma(value):
          value=str(value)
          if ',' in value:
              value=value.replace(',','')
              return float(value)
          else:
              return float(value)

          df['CostFor2Plates']=df['CostFor2Plates'].apply(handlecomma)
          df['CostFor2Plates'].unique()
```

```
Out[20]: array([ 800., 300., 600., 700., 550., 500., 450., 650., 400.,
          900., 200., 750., 150., 850., 100., 1200., 350., 250.,
          950., 1000., 1500., 1300., 199., 80., 1100., 160., 1600.,
          230., 130., 50., 190., 1700., 1400., 180., 1350., 2200.,
          2000., 1800., 1900., 330., 2500., 2100., 3000., 2800., 3400.,
          40., 1250., 3500., 4000., 2400., 2600., 120., 1450., 469.,
          70., 3200., 60., 560., 240., 360., 6000., 1050., 2300.,
          4100., 5000., 3700., 1650., 2700., 4500., 140.])
```

```
In [21]: df.head()
```

```
Out[21]:
```

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	CostFor2Plates	Type
0	Jalsa	Yes	Yes	4.1	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	800.0	Buffet
1	Spice Elephant	Yes	No	4.1	787	Banashankari	Casual Dining	Chinese, North Indian, Thai	800.0	Buffet
2	San Churro Cafe	Yes	No	3.8	918	Banashankari	Cafe, Casual Dining	Cafe, Mexican, Italian	800.0	Buffet
3	Addhuri Udupi Bhojana	No	No	3.7	88	Banashankari	Quick Bites	South Indian, North Indian	300.0	Buffet
4	Grand Village	No	No	3.8	166	Basavanagudi	Casual Dining	North Indian, Rajasthani	600.0	Buffet

Cleaning Rest Type Column

```
In [22]: rest_types=df['rest_type'].value_counts(ascending=False)
          rest_types
```

```
Out[22]: rest_type
Quick Bites                19010
Casual Dining              10253
Cafe                       3682
Delivery                   2574
Dessert Parlor             2242
...
Dessert Parlor, Kiosk       2
Food Court, Beverage Shop   2
Dessert Parlor, Food Court  2
Quick Bites, Kiosk          1
Sweet Shop, Dessert Parlor   1
Name: count, Length: 93, dtype: int64
```

```
In [23]: rest_types_lessThan1000=rest_types[rest_types<1000]
          rest_types_lessThan1000
```

```
Out[23]: rest_type
Beverage Shop              863
Bar                        686
Food Court                  616
Sweet Shop                  468
Bar, Casual Dining          411
...
Dessert Parlor, Kiosk       2
Food Court, Beverage Shop   2
Dessert Parlor, Food Court  2
Quick Bites, Kiosk          1
Sweet Shop, Dessert Parlor   1
Name: count, Length: 85, dtype: int64
```

Making rest_types_lessThan1000 in frequency as Other

```
In [24]: def handle_rest_type(value):
         if(value in rest_types_lessThan1000):
             return "others"
         else:
             return value

         df['rest_type']=df['rest_type'].apply(handle_rest_type)
         df['rest_type'].value_counts()
```

```
Out[24]: rest_type
Quick Bites      19010
Casual Dining    10253
others           9003
Cafe             3682
Delivery         2574
Dessert Parlor   2242
Takeaway, Delivery 2008
Bakery           1140
Casual Dining, Bar 1130
Name: count, dtype: int64
```

```
In [25]: df.head(10)
```

```
Out[25]:
```

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	CostFor2Plates	Type
0	Jalsa	Yes	Yes	4.1	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	800.0	Buffet
1	Spice Elephant	Yes	No	4.1	787	Banashankari	Casual Dining	Chinese, North Indian, Thai	800.0	Buffet
2	San Churro Cafe	Yes	No	3.8	918	Banashankari	others	Cafe, Mexican, Italian	800.0	Buffet
3	Addhuri Udupi Bhojana	No	No	3.7	88	Banashankari	Quick Bites	South Indian, North Indian	300.0	Buffet
4	Grand Village	No	No	3.8	166	Basavanagudi	Casual Dining	North Indian, Rajasthani	600.0	Buffet
5	Timepass Dinner	Yes	No	3.8	286	Basavanagudi	Casual Dining	North Indian	600.0	Buffet
6	Rosewood International Hotel - Bar & Restaurant	No	No	3.6	8	Mysore Road	Casual Dining	North Indian, South Indian, Andhra, Chinese	800.0	Buffet
7	Onesta	Yes	Yes	4.6	2556	Banashankari	others	Pizza, Cafe, Italian	600.0	Cafes
8	Penthouse Cafe	Yes	No	4.0	324	Banashankari	Cafe	Cafe, Italian, Continental	700.0	Cafes
9	Smacznego	Yes	No	4.2	504	Banashankari	Cafe	Cafe, Mexican, Italian, Momos, Beverages	550.0	Cafes

Making location less than 300 in frequency as other

```
In [26]: df['location'].value_counts()
```

```
Out[26]: location
BTM      5056
HSR      2494
Koramangala 5th Block 2479
JP Nagar  2218
Whitefield 2105
...
West Bangalore      6
Yelahanka           5
Jakkur              3
Rajarajeshwari Nagar 2
Peenya              1
Name: count, Length: 93, dtype: int64
```

```
In [27]: location=df['location'].value_counts(ascending=False)
location_lessThan300=location[location<300]

def handel_location(value):
    if(value in location_lessThan300):
        return "other"
    else:
        return value

df['location']=df['location'].apply(handel_location)
df['location'].value_counts()
```

```
Out[27]: location
BTM                    5056
other                  4954
HSR                    2494
Koramangala 5th Block  2479
JP Nagar               2218
Whitefield             2105
Indiranagar            2026
Jayanagar              1916
Marathahalli           1805
Bannerghatta Road     1609
Bellandur              1268
Electronic City        1246
Koramangala 1st Block  1236
Brigade Road           1210
Koramangala 7th Block  1174
Koramangala 6th Block  1127
Sarjapur Road          1047
Koramangala 4th Block  1017
Ulsoor                 1011
Banashankari           902
MG Road                893
Kalyan Nagar           841
Richmond Road          803
Malleshwaram           721
Frazer Town            714
Basavanagudi           684
Residency Road         671
Brookefield            656
New BEL Road           644
Banaswadi              640
Kammanahalli           639
Rajajinagar            591
Church Street          566
Lavelle Road           518
Shanti Nagar           508
Shivajinagar           498
Cunningham Road        490
Domlur                 482
Old Airport Road       437
Ejipura                433
Commercial Street      370
St. Marks Road         343
Name: count, dtype: int64
```

```
In [28]: df.head(10)
```

```
Out[28]:
```

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	CostFor2Plates	Type
0	Jalsa	Yes	Yes	4.1	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	800.0	Buffet
1	Spice Elephant	Yes	No	4.1	787	Banashankari	Casual Dining	Chinese, North Indian, Thai	800.0	Buffet
2	San Churro Cafe	Yes	No	3.8	918	Banashankari	others	Cafe, Mexican, Italian	800.0	Buffet
3	Addhuri Udupi Bhojana	No	No	3.7	88	Banashankari	Quick Bites	South Indian, North Indian	300.0	Buffet
4	Grand Village	No	No	3.8	166	Basavanagudi	Casual Dining	North Indian, Rajasthani	600.0	Buffet
5	Timepass Dinner	Yes	No	3.8	286	Basavanagudi	Casual Dining	North Indian	600.0	Buffet
6	Rosewood International Hotel - Bar & Restaurant	No	No	3.6	8	other	Casual Dining	North Indian, South Indian, Andhra, Chinese	800.0	Buffet
7	Onesta	Yes	Yes	4.6	2556	Banashankari	others	Pizza, Cafe, Italian	600.0	Cafes
8	Penthouse Cafe	Yes	No	4.0	324	Banashankari	Cafe	Cafe, Italian, Continental	700.0	Cafes
9	Smaczego	Yes	No	4.2	504	Banashankari	Cafe	Cafe, Mexican, Italian, Momos, Beverages	550.0	Cafes

Making cuisines less than 100 in frequency as other

```
In [29]: cuisines=df['cuisines'].value_counts(ascending=False)
cuisines_lessThan100= cuisines[cuisines<100]

def handle_Cuisines(value):
    if value in cuisines_lessThan100:
        return "other"
    else:
        return value

df['cuisines']=df['cuisines'].apply(handle_Cuisines)
df['cuisines'].value_counts()
```

```
Out[29]: cuisines
other                26159
North Indian         2852
North Indian, Chinese 2351
South Indian         1820
Biryani              903
...
South Indian, Chinese, North Indian 105
North Indian, Mughlai, Chinese      104
South Indian, Fast Food             104
Italian, Pizza                     102
North Indian, Chinese, Seafood      102
Name: count, Length: 70, dtype: int64
```

```
In [30]: df.head()
```

```
Out[30]:
```

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	CostFor2Plates	Type
0	Jalsa	Yes	Yes	4.1	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	800.0	Buffet
1	Spice Elephant	Yes	No	4.1	787	Banashankari	Casual Dining	other	800.0	Buffet
2	San Churro Cafe	Yes	No	3.8	918	Banashankari	others	other	800.0	Buffet
3	Addhuri Udupi Bhojana	No	No	3.7	88	Banashankari	Quick Bites	South Indian, North Indian	300.0	Buffet
4	Grand Village	No	No	3.8	166	Basavanagudi	Casual Dining	other	600.0	Buffet

```
In [31]: df['Type'].value_counts()
```

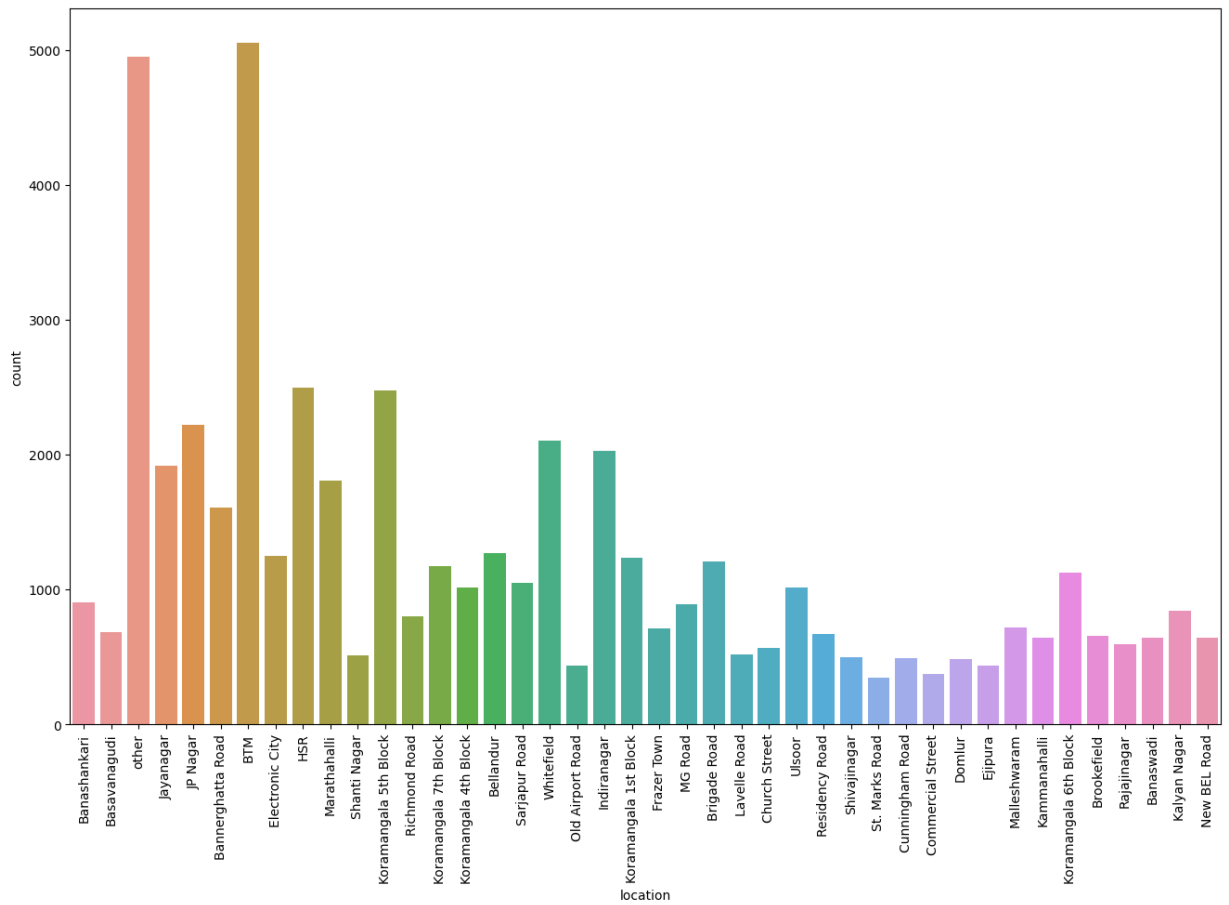
```
Out[31]: Type
Delivery                25579
Dine-out                17562
Desserts                 3559
Cafes                   1703
Drinks & nightlife      1084
Buffet                  869
Pubs and bars           686
Name: count, dtype: int64
```


Data is cleaned and now we can make the visualization

Count Plot of various Locations

```
In [32]: plt.figure(figsize=(16, 10))
ax = sns.countplot(data=df, x='location')
plt.xticks(rotation=90)
```

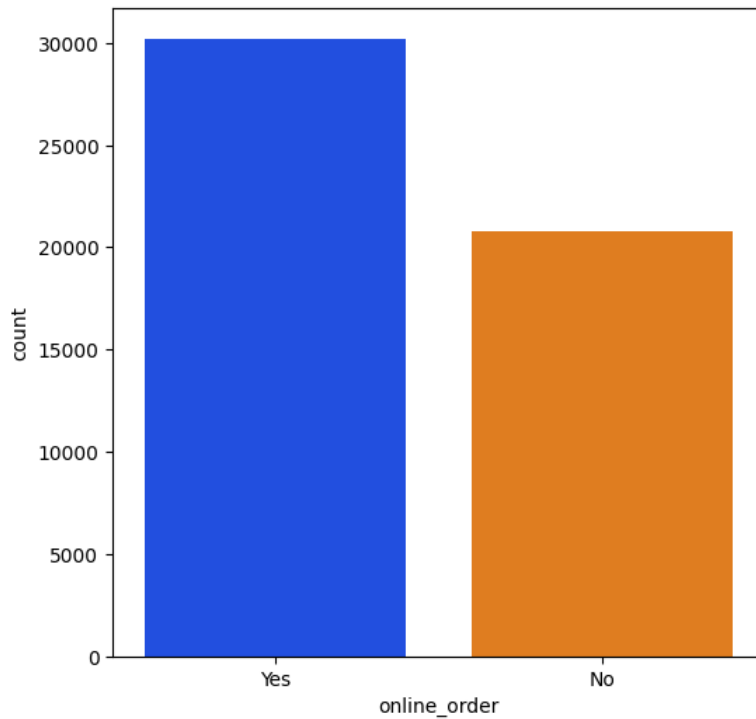
```
Out[32]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
        34, 35, 36, 37, 38, 39, 40, 41]),
 [Text(0, 0, 'Banashankari'),
 Text(1, 0, 'Basavanagudi'),
 Text(2, 0, 'other'),
 Text(3, 0, 'Jayanagar'),
 Text(4, 0, 'JP Nagar'),
 Text(5, 0, 'Bannerghatta Road'),
 Text(6, 0, 'BTM'),
 Text(7, 0, 'Electronic City'),
 Text(8, 0, 'HSR'),
 Text(9, 0, 'Marathahalli'),
 Text(10, 0, 'Shanti Nagar'),
 Text(11, 0, 'Koramangala 5th Block'),
 Text(12, 0, 'Richmond Road'),
 Text(13, 0, 'Koramangala 7th Block'),
 Text(14, 0, 'Koramangala 4th Block'),
 Text(15, 0, 'Bellandur'),
 Text(16, 0, 'Sarjapur Road'),
 Text(17, 0, 'Whitefield'),
 Text(18, 0, 'Old Airport Road'),
 Text(19, 0, 'Indiranagar'),
 Text(20, 0, 'Koramangala 1st Block'),
 Text(21, 0, 'Frazer Town'),
 Text(22, 0, 'MG Road'),
 Text(23, 0, 'Brigade Road'),
 Text(24, 0, 'Lavelle Road'),
 Text(25, 0, 'Church Street'),
 Text(26, 0, 'Ulsoor'),
 Text(27, 0, 'Residency Road'),
 Text(28, 0, 'Shivajinagar'),
 Text(29, 0, 'St. Marks Road'),
 Text(30, 0, 'Cunningham Road'),
 Text(31, 0, 'Commercial Street'),
 Text(32, 0, 'Domlur'),
 Text(33, 0, 'Ejipura'),
 Text(34, 0, 'Malleshwaram'),
 Text(35, 0, 'Kammanahalli'),
 Text(36, 0, 'Koramangala 6th Block'),
 Text(37, 0, 'Brookefield'),
 Text(38, 0, 'Rajajinagar'),
 Text(39, 0, 'Banaswadi'),
 Text(40, 0, 'Kalyan Nagar'),
 Text(41, 0, 'New BEL Road')])
```



Visualizing Online order

```
In [33]: plt.figure(figsize=(6, 6))
sns.countplot(data=df, x='online_order', palette='bright')
```

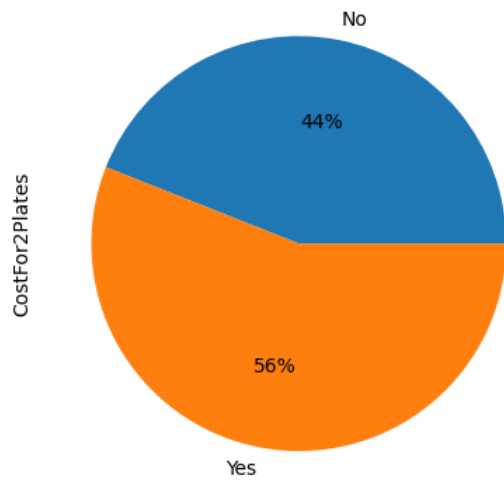
```
Out[33]: <Axes: xlabel='online_order', ylabel='count'>
```



Visualizing Sale v/s Online order preferences

```
In [34]: df.groupby('online_order')['CostFor2Plates'].sum().plot(kind='pie', autopct="%1.0f%%")
```

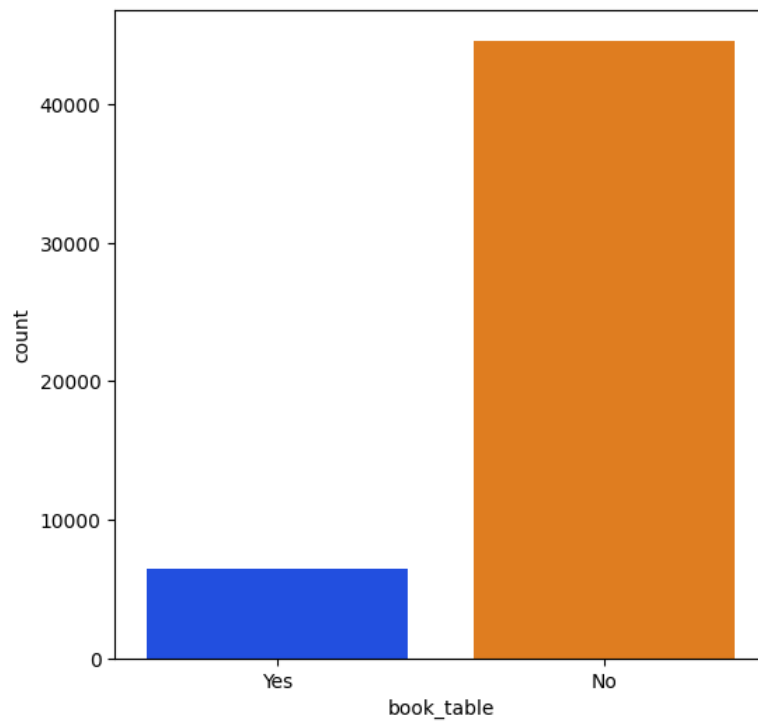
```
Out[34]: <Axes: ylabel='CostFor2Plates'>
```



Visualizing Book Table

```
In [35]: plt.figure(figsize=(6, 6))  
sns.countplot(data=df, x='book_table', palette='bright')
```

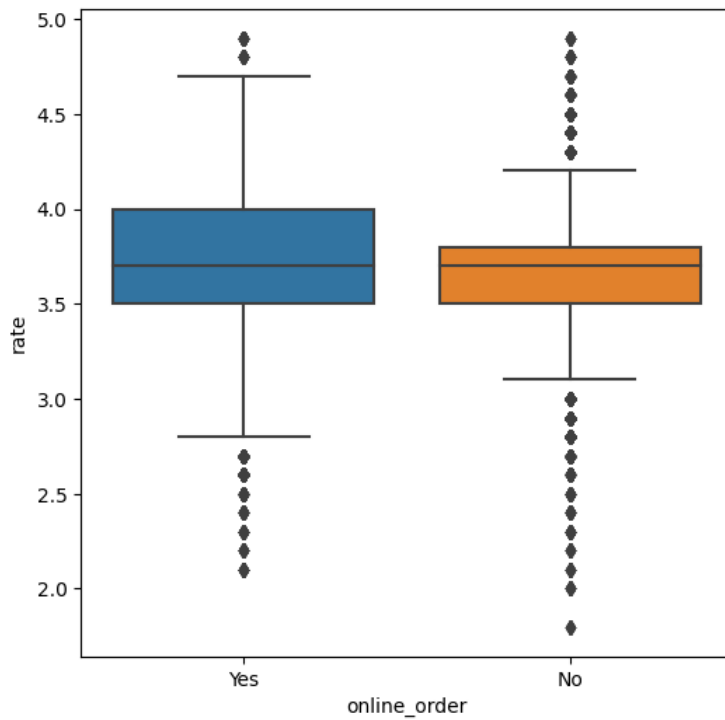
```
Out[35]: <Axes: xlabel='book_table', ylabel='count'>
```



Visualization of Online Order vs Rate

```
In [36]: plt.figure(figsize=(6,6))  
sns.boxplot(x='online_order', y='rate',data=df)
```

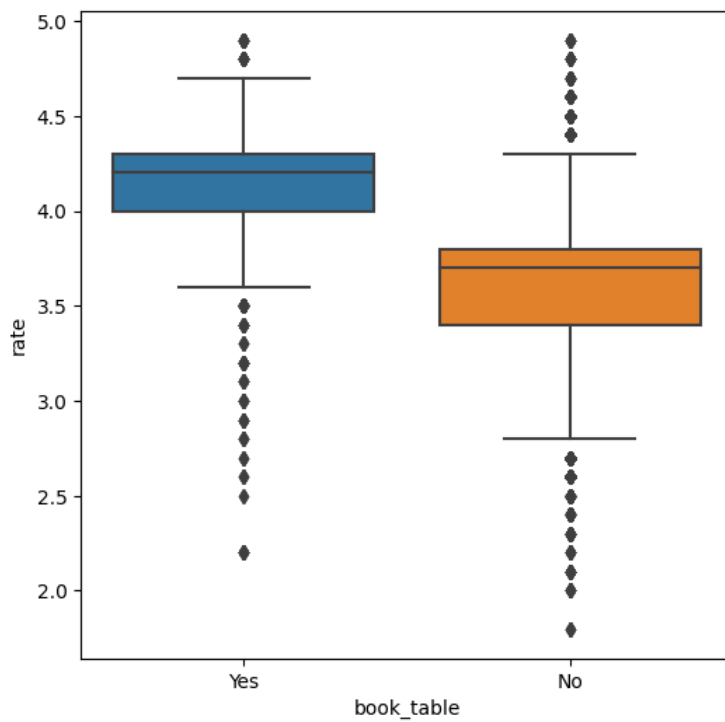
```
Out[36]: <Axes: xlabel='online_order', ylabel='rate'>
```



Visualization of Book Table vs Rate

```
In [37]: plt.figure(figsize=(6,6))  
sns.boxplot(x='book_table', y='rate',data=df)
```

```
Out[37]: <Axes: xlabel='book_table', ylabel='rate'>
```



Visualization of Online Order Facility, Location wise

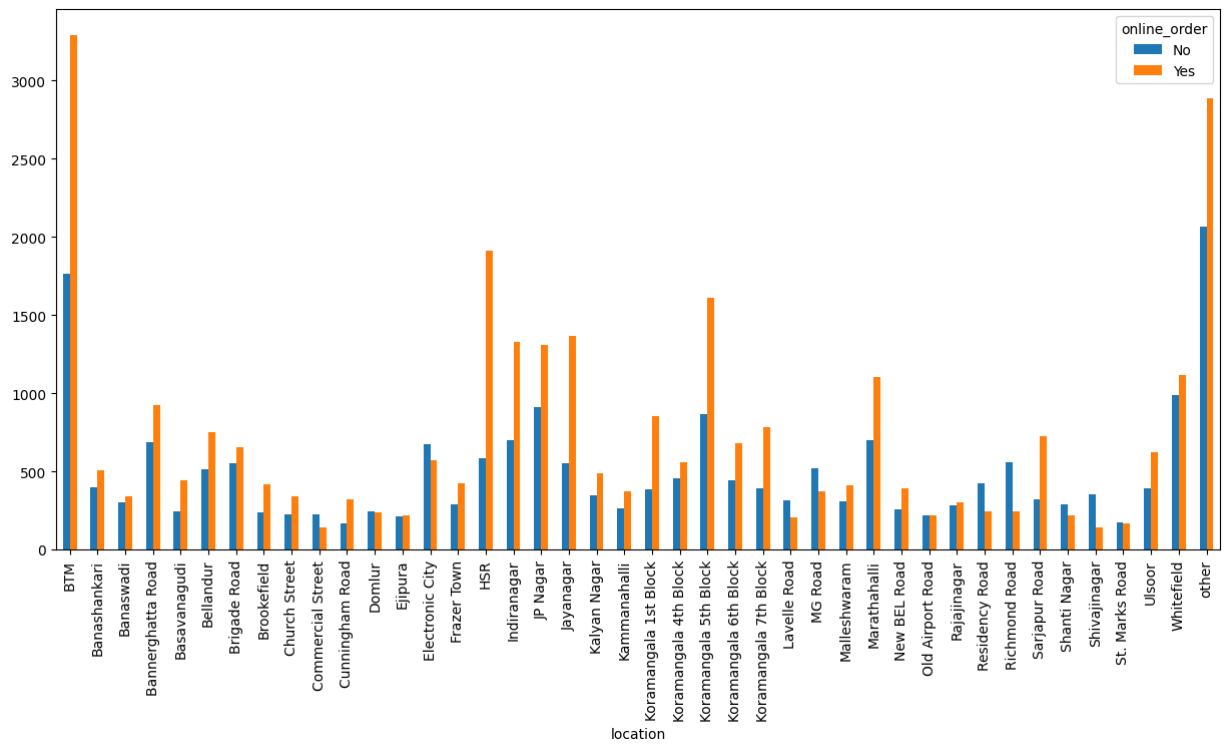
```
In [38]: df1 = df.groupby(['location', 'online_order'])['name'].count()
df1.to_csv('PivotFile.csv')
df1 = pd.read_csv('PivotFile.csv')
df1 = pd.pivot_table(df1, values='name', index='location', columns='online_order', fill_value=0, aggfunc=np.sum)
df1
```

Out[38]:

online_order	No	Yes
location		
BTM	1763	3293
Banashankari	397	505
Banaswadi	302	338
Bannerghatta Road	685	924
Basavanagudi	243	441
Bellandur	517	751
Brigade Road	552	658
Brookefield	239	417
Church Street	226	340
Commercial Street	228	142
Cunningham Road	168	322
Domlur	247	235
Ejipura	214	219
Electronic City	676	570
Frazer Town	287	427
HSR	584	1910
Indiranagar	697	1329
JP Nagar	911	1307
Jayanagar	552	1364
Kalyan Nagar	350	491
Kammanahalli	264	375
Koramangala 1st Block	384	852
Koramangala 4th Block	459	558
Koramangala 5th Block	866	1613
Koramangala 6th Block	445	682
Koramangala 7th Block	389	785
Lavelle Road	315	203
MG Road	520	373
Malleshwaram	309	412
Marathahalli	701	1104
New BEL Road	255	389
Old Airport Road	221	216
Rajajinagar	286	305
Residency Road	424	247
Richmond Road	557	246
Sarjapur Road	323	724
Shanti Nagar	289	219
Shivajinagar	354	144
St. Marks Road	176	167
Ulsoor	389	622
Whitefield	986	1119
other	2064	2890

```
In [39]: df1.plot(kind='bar',figsize=(15,7))
```

```
Out[39]: <Axes: xlabel='location'>
```



Visualizing Book Table Facility, Location Wise

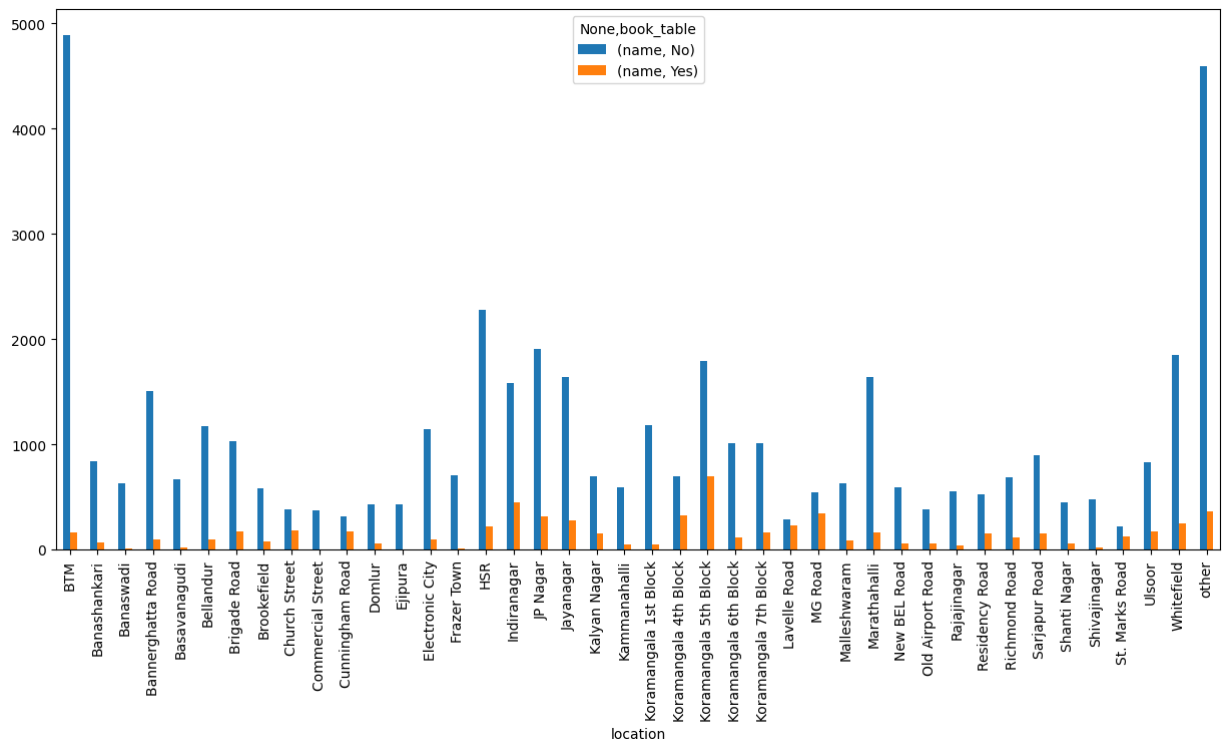
```
In [40]: df2=df.groupby(['location','book_table'])['name'].count()
df2.to_csv('Location_bookTable.csv')
df2=pd.read_csv('Location_bookTable.csv')
df2=pd.pivot_table(df2,values=None,index=['location'],columns=['book_table'],fill_value=0,aggfunc=np.sum)
df2
```

Out[40]:

book_table	name	
	No	Yes
location		
BTM	4889	167
Banashankari	839	63
Banaswadi	632	8
Bannerghatta Road	1510	99
Basavanagudi	668	16
Bellandur	1170	98
Brigade Road	1034	176
Brookefield	582	74
Church Street	385	181
Commercial Street	370	0
Cunningham Road	315	175
Domlur	427	55
Ejipura	433	0
Electronic City	1148	98
Frazer Town	706	8
HSR	2277	217
Indiranagar	1578	448
JP Nagar	1903	315
Jayanagar	1637	279
Kalyan Nagar	692	149
Kammanahalli	590	49
Koramangala 1st Block	1186	50
Koramangala 4th Block	695	322
Koramangala 5th Block	1787	692
Koramangala 6th Block	1015	112
Koramangala 7th Block	1012	162
Lavelle Road	290	228
MG Road	546	347
Malleshwaram	632	89
Marathahalli	1642	163
New BEL Road	588	56
Old Airport Road	378	59
Rajajinagar	550	41
Residency Road	522	149
Richmond Road	687	116
Sarjapur Road	893	154
Shanti Nagar	451	57
Shivajinagar	475	23
St. Marks Road	219	124
Ulsoor	834	177
Whitefield	1852	253
other	4587	367

```
In [41]: df2.plot(kind='bar',figsize=(15,7))
```

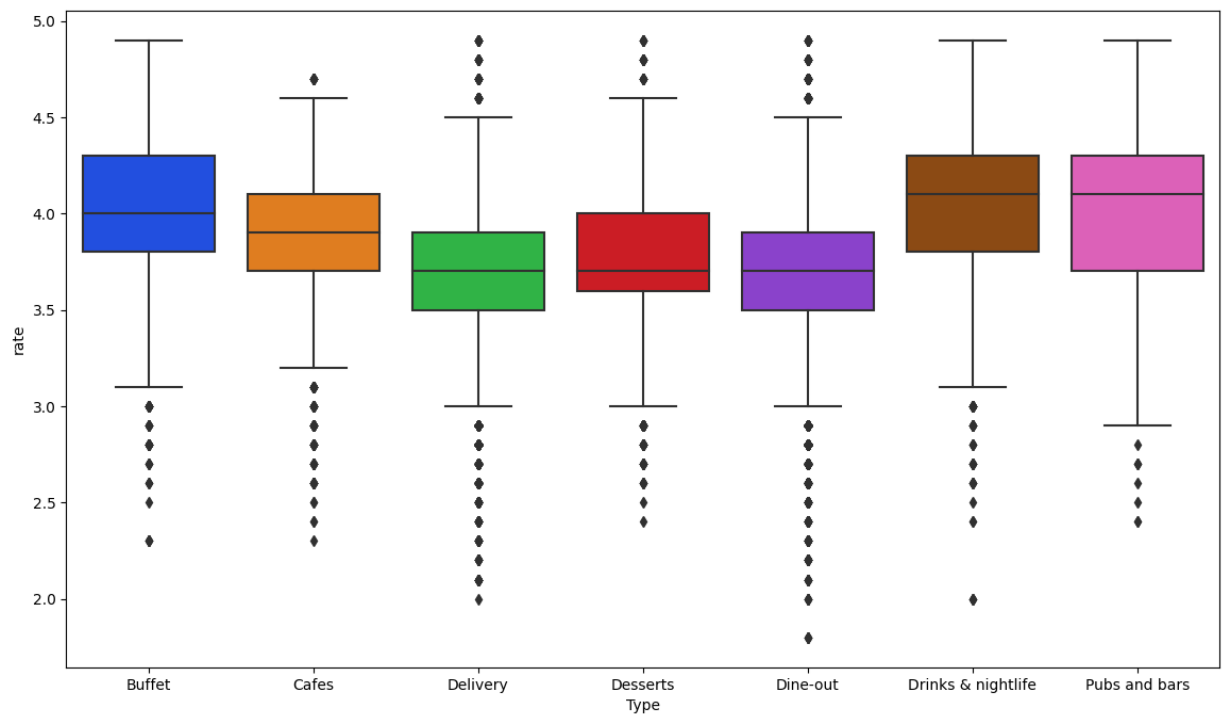
```
Out[41]: <Axes: xlabel='location'>
```



Visualizing Types of Restaurants vs Rate

```
In [42]: plt.figure(figsize=(14,8))
sns.boxplot(x='Type',y='rate',data=df,palette='bright')
```

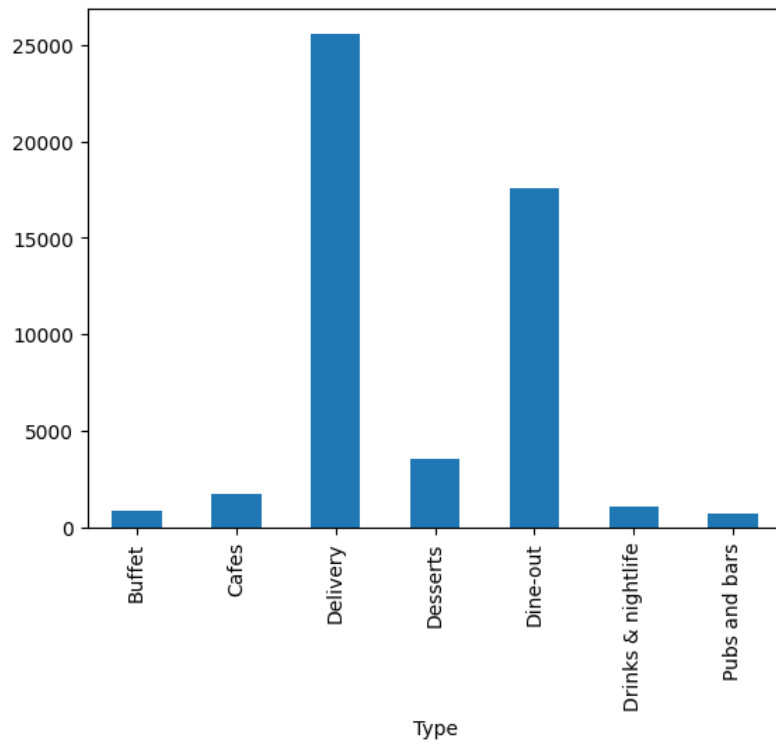
```
Out[42]: <Axes: xlabel='Type', ylabel='rate'>
```



Visualizing Type v/s Votes

```
In [43]: df.groupby('Type')['votes'].count().plot(kind='bar')
```

```
Out[43]: <Axes: xlabel='Type'>
```



Grouping Types of Restaurents, Location wise

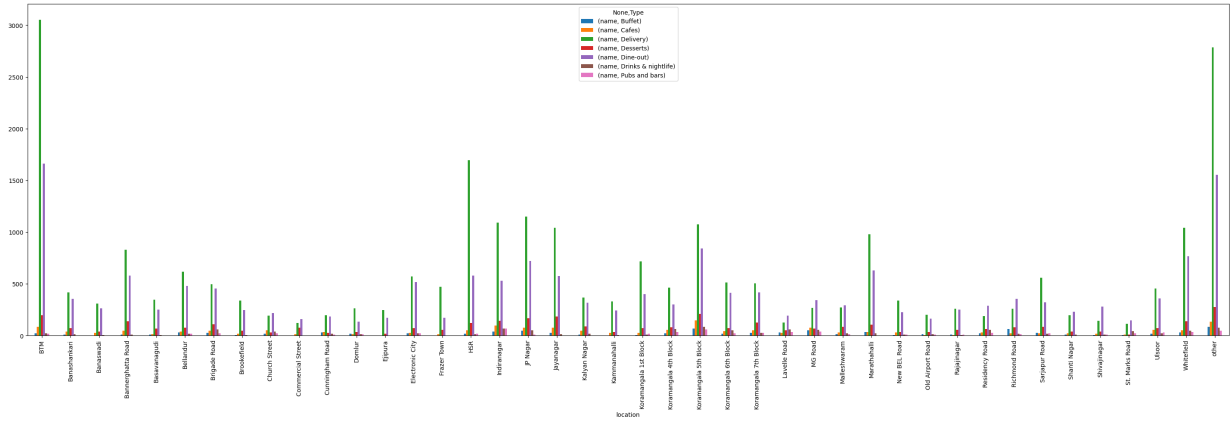
```
In [44]: df3=df.groupby(['location','Type'])['name'].count()
df3.to_csv('location_Type.csv')
df3=pd.read_csv('location_Type.csv')
df3=pd.pivot_table(df3,values=None,index=['location'],columns=['Type'],fill_value=0,aggfunc=np.sum)
df3
```

Out[44]:

	name							
Type	Buffet	Cafes	Delivery	Desserts	Dine-out	Drinks & nightlife	Pubs and bars	
location								
BTM	21	83	3053	198	1660	22	19	
Banashankari	7	36	418	71	356	14	0	
Banaswadi	0	24	310	37	262	6	1	
Bannerghatta Road	9	46	828	137	578	9	2	
Basavanagudi	7	11	344	66	251	5	0	
Bellandur	28	36	617	75	479	17	16	
Brigade Road	25	46	497	108	455	57	22	
Brookefield	6	17	339	45	245	4	0	
Church Street	19	51	193	29	215	36	23	
Commercial Street	0	13	121	77	159	0	0	
Cunningham Road	29	34	194	26	184	16	7	
Domlur	15	13	261	35	135	12	11	
Ejipura	0	0	245	16	172	0	0	
Electronic City	23	24	570	71	516	21	21	
Frazer Town	1	11	470	56	172	2	2	
HSR	19	49	1694	120	580	14	18	
Indiranagar	38	97	1091	140	529	65	66	
JP Nagar	45	76	1151	166	722	51	7	
Jayanagar	27	77	1043	182	575	12	0	
Kalyan Nagar	9	45	366	88	315	18	0	
Kammanahalli	2	27	329	35	240	6	0	
Koramangala 1st Block	3	26	716	70	398	7	16	
Koramangala 4th Block	21	53	464	81	302	62	34	
Koramangala 5th Block	65	146	1075	209	842	84	58	
Koramangala 6th Block	18	43	511	70	411	51	23	
Koramangala 7th Block	25	52	503	127	417	25	25	
Lavelle Road	30	27	127	50	191	59	34	
MG Road	51	76	266	68	343	53	36	
Malleshwaram	11	31	269	85	291	20	14	
Marathahalli	34	32	980	105	630	22	2	
New BEL Road	4	29	338	33	224	8	8	
Old Airport Road	12	5	200	35	164	12	9	
Rajajinagar	10	4	258	55	251	3	10	
Residency Road	20	31	187	63	289	55	26	
Richmond Road	63	21	257	78	356	16	12	
Sarjapur Road	25	22	558	82	319	19	22	
Shanti Nagar	9	22	198	39	229	9	2	
Shivajinagar	6	17	143	37	280	7	8	
St. Marks Road	5	10	111	10	145	40	22	
Ulsoor	16	56	456	71	359	23	30	
Whitefield	28	51	1041	137	768	47	33	
other	83	133	2787	276	1553	75	47	

```
In [45]: df3.plot(kind='bar',figsize=(36,10))
```

```
Out[45]: <Axes: xlabel='location'>
```



Number of Votes, Loation wise

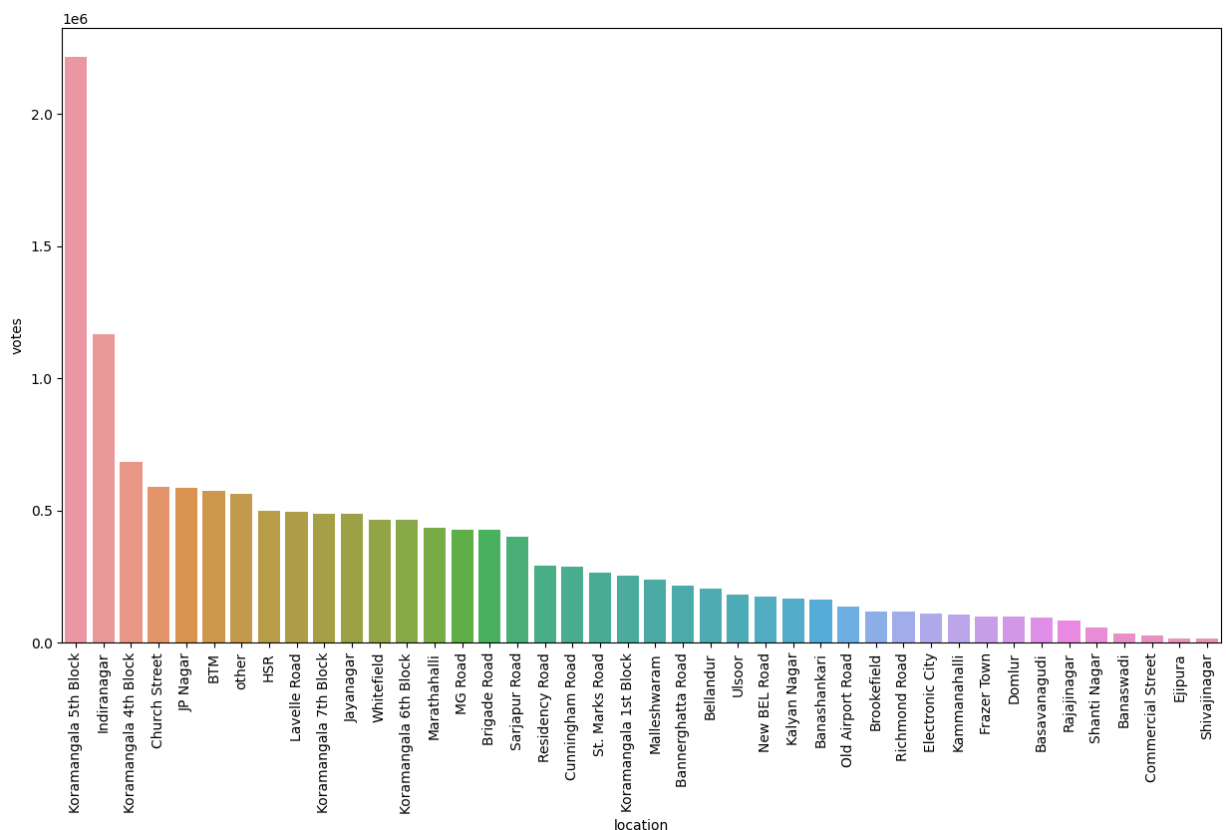
```
In [46]: df4=df[['location','votes']]
df4.drop_duplicates()
df5=df4.groupby(['location'])['votes'].sum()
df5=df5.to_frame()
df5=df5.sort_values('votes',ascending=False)
df5.head()
```

```
Out[46]:
```

votes	
location	
Koramangala 5th Block	2214083
Indiranagar	1165909
Koramangala 4th Block	685156
Church Street	590306
JP Nagar	586522

```
In [47]: plt.figure(figsize=(15, 8))
sns.barplot(x=df5.index, y=df5['votes'])
plt.xticks(rotation=90)
```

```
Out[47]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
        34, 35, 36, 37, 38, 39, 40, 41]),
 [Text(0, 0, 'Koramangala 5th Block'),
 Text(1, 0, 'Indiranagar'),
 Text(2, 0, 'Koramangala 4th Block'),
 Text(3, 0, 'Church Street'),
 Text(4, 0, 'JP Nagar'),
 Text(5, 0, 'BTM'),
 Text(6, 0, 'other'),
 Text(7, 0, 'HSR'),
 Text(8, 0, 'Lavelle Road'),
 Text(9, 0, 'Koramangala 7th Block'),
 Text(10, 0, 'Jayanagar'),
 Text(11, 0, 'Whitefield'),
 Text(12, 0, 'Koramangala 6th Block'),
 Text(13, 0, 'Marathahalli'),
 Text(14, 0, 'MG Road'),
 Text(15, 0, 'Brigade Road'),
 Text(16, 0, 'Sarjapur Road'),
 Text(17, 0, 'Residency Road'),
 Text(18, 0, 'Cunningham Road'),
 Text(19, 0, 'St. Marks Road'),
 Text(20, 0, 'Koramangala 1st Block'),
 Text(21, 0, 'Malleshwaram'),
 Text(22, 0, 'Bannerghatta Road'),
 Text(23, 0, 'Bellandur'),
 Text(24, 0, 'Ulsoor'),
 Text(25, 0, 'New BEL Road'),
 Text(26, 0, 'Kalyan Nagar'),
 Text(27, 0, 'Banashankari'),
 Text(28, 0, 'Old Airport Road'),
 Text(29, 0, 'Brookefield'),
 Text(30, 0, 'Richmond Road'),
 Text(31, 0, 'Electronic City'),
 Text(32, 0, 'Kammanahalli'),
 Text(33, 0, 'Frazer Town'),
 Text(34, 0, 'Domlur'),
 Text(35, 0, 'Basavanagudi'),
 Text(36, 0, 'Rajajinagar'),
 Text(37, 0, 'Shanti Nagar'),
 Text(38, 0, 'Banaswadi'),
 Text(39, 0, 'Commercial Street'),
 Text(40, 0, 'Ejipura'),
 Text(41, 0, 'Shivajinagar')])
```



Visualizing Top Cuisines

```
In [48]: df6=df[['cuisines','votes']]
df6.drop_duplicates()
df7=df6.groupby(['cuisines'])['votes'].sum()
df7=df7.to_frame()
df7=df7.sort_values('votes',ascending=False)
df7.head()
```

Out[48]:

votes	
cuisines	
other	11542182
North Indian	516310
North Indian, Chinese	258225
South Indian	161975
North Indian, Mughlai	103706

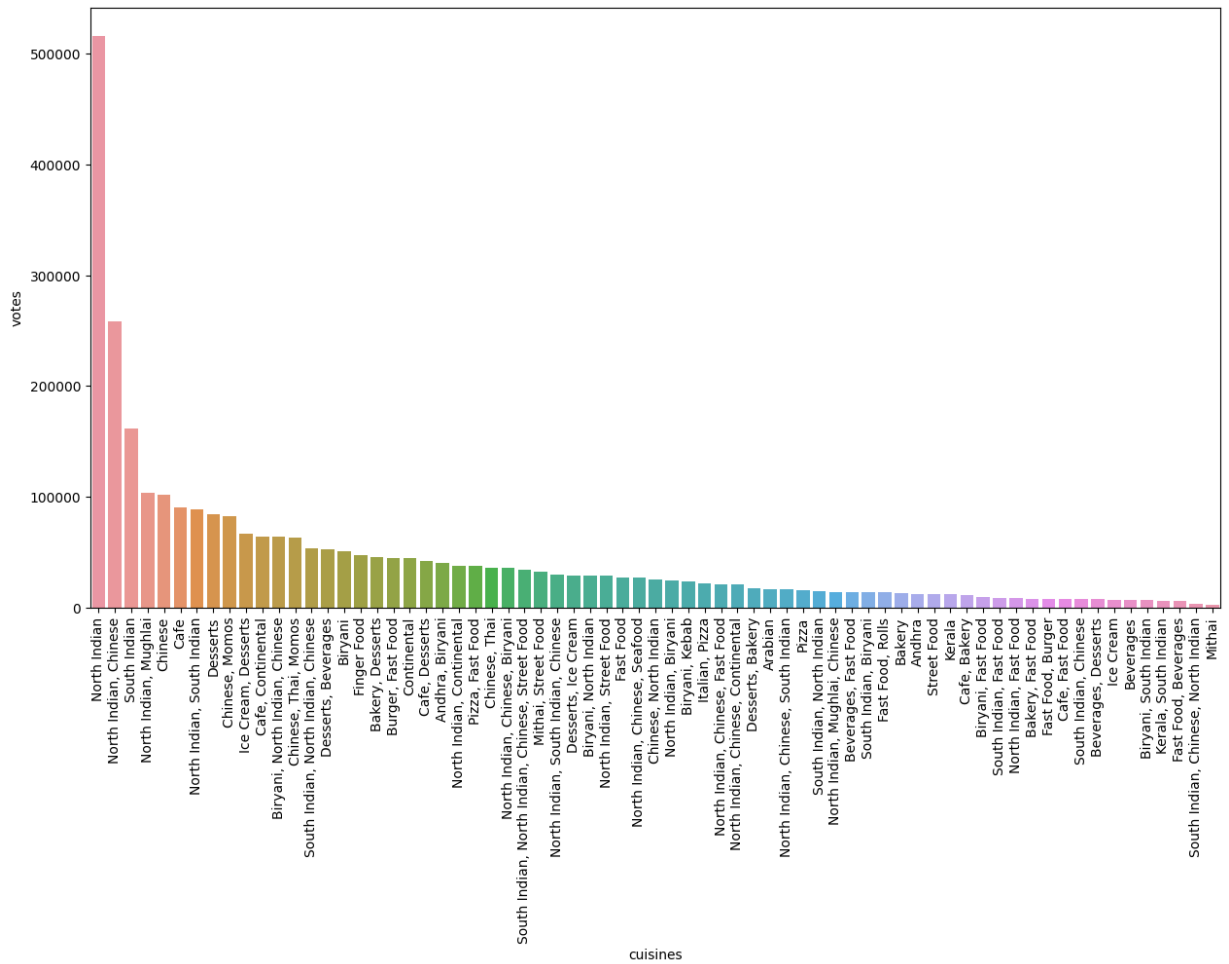
```
In [49]: df7=df7.iloc[1:,:]
df7.head()
```

Out[49]:

votes	
cuisines	
North Indian	516310
North Indian, Chinese	258225
South Indian	161975
North Indian, Mughlai	103706
Chinese	101728

```
In [50]: plt.figure(figsize=(15,8))
sns.barplot(x=df7.index,y=df7['votes'])
plt.xticks(rotation=90)
```

```
Out[50]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
        34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
        51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67,
        68]),
[Text(0, 0, 'North Indian'),
Text(1, 0, 'North Indian, Chinese'),
Text(2, 0, 'South Indian'),
Text(3, 0, 'North Indian, Mughlai'),
Text(4, 0, 'Chinese'),
Text(5, 0, 'Cafe'),
Text(6, 0, 'North Indian, South Indian'),
Text(7, 0, 'Desserts'),
Text(8, 0, 'Chinese, Momos'),
Text(9, 0, 'Ice Cream, Desserts'),
Text(10, 0, 'Cafe, Continental'),
Text(11, 0, 'Biryani, North Indian, Chinese'),
Text(12, 0, 'Chinese, Thai, Momos'),
Text(13, 0, 'South Indian, North Indian, Chinese'),
Text(14, 0, 'Desserts, Beverages'),
Text(15, 0, 'Biryani'),
Text(16, 0, 'Finger Food'),
Text(17, 0, 'Bakery, Desserts'),
Text(18, 0, 'Burger, Fast Food'),
Text(19, 0, 'Continental'),
Text(20, 0, 'Cafe, Desserts'),
Text(21, 0, 'Andhra, Biryani'),
Text(22, 0, 'North Indian, Continental'),
Text(23, 0, 'Pizza, Fast Food'),
Text(24, 0, 'Chinese, Thai'),
Text(25, 0, 'North Indian, Chinese, Biryani'),
Text(26, 0, 'South Indian, North Indian, Chinese, Street Food'),
Text(27, 0, 'Mithai, Street Food'),
Text(28, 0, 'North Indian, South Indian, Chinese'),
Text(29, 0, 'Desserts, Ice Cream'),
Text(30, 0, 'Biryani, North Indian'),
Text(31, 0, 'North Indian, Street Food'),
Text(32, 0, 'Fast Food'),
Text(33, 0, 'North Indian, Chinese, Seafood'),
Text(34, 0, 'Chinese, North Indian'),
Text(35, 0, 'North Indian, Biryani'),
Text(36, 0, 'Biryani, Kebab'),
Text(37, 0, 'Italian, Pizza'),
Text(38, 0, 'North Indian, Chinese, Fast Food'),
Text(39, 0, 'North Indian, Chinese, Continental'),
Text(40, 0, 'Desserts, Bakery'),
Text(41, 0, 'Arabian'),
Text(42, 0, 'North Indian, Chinese, South Indian'),
Text(43, 0, 'Pizza'),
Text(44, 0, 'South Indian, North Indian'),
Text(45, 0, 'North Indian, Mughlai, Chinese'),
Text(46, 0, 'Beverages, Fast Food'),
Text(47, 0, 'South Indian, Biryani'),
Text(48, 0, 'Fast Food, Rolls'),
Text(49, 0, 'Bakery'),
Text(50, 0, 'Andhra'),
Text(51, 0, 'Street Food'),
Text(52, 0, 'Kerala'),
Text(53, 0, 'Cafe, Bakery'),
Text(54, 0, 'Biryani, Fast Food'),
Text(55, 0, 'South Indian, Fast Food'),
Text(56, 0, 'North Indian, Fast Food'),
Text(57, 0, 'Bakery, Fast Food'),
Text(58, 0, 'Fast Food, Burger'),
Text(59, 0, 'Cafe, Fast Food'),
Text(60, 0, 'South Indian, Chinese'),
Text(61, 0, 'Beverages, Desserts'),
Text(62, 0, 'Ice Cream'),
Text(63, 0, 'Beverages'),
Text(64, 0, 'Biryani, South Indian'),
Text(65, 0, 'Kerala, South Indian'),
Text(66, 0, 'Fast Food, Beverages'),
Text(67, 0, 'South Indian, Chinese, North Indian'),
Text(68, 0, 'Mithai')])
```



In []:

In []:

In []: